

1. (Adaptado de Goodrich e Tamassia R-23.1) Seja $P = aaabbaaa$. Qual o tamanho do conjunto $\{k > 0 \mid P_k \sqsupseteq P\}$? Qual o de $\sigma(ccaabbaa)$? Compute a função de transição para o AFD para o casamento desse padrão.
2. Construa o autômato que represente o padrão 0110200100 e mostre o funcionamento do algoritmo para casamento de padrões baseado em AFD com o texto 00202320101302001000020232010110200100.
3. (CLRS 32.3-5) Suponha que seja permitida a utilização de *curingas* (símbolos que casam com qualquer outro) na construção do padrão. Mostre como adaptar o algoritmo baseado em autômatos para continuar a encontrar o padrão em tempo $O(n)$.
4. (Goodrich e Tamassia R-23.7) Compute a função prefixo ($\pi(q)$) para o padrão *cgtacgttcgtac*.
5. (Goodrich e Tamassia R-23.5) Compute a função *shift* do algoritmo de Horspool para o padrão “the quick brown fox jumped over a lazy cat”.
6. (Levitin 7.3-3) Quantas comparações serão realizadas pelo algoritmo de Horspool para encontrar os padrões a seguir no texto 0^{1000} .
 - (a) 00001
 - (b) 10000
 - (c) 01010
7. Aponte qual a melhor escolha de algoritmo para casamento de padrões em texto considerando cada situação:
 - (a) Deseja-se verificar se algum código malicioso contido em uma lista pré-definida ocorre em um arquivo recebido por email.
 - (b) Deseja-se filtrar posts no Twitter que possuam uma palavra-chave definida pelo usuário.
8. (Goodrich e Tamassia R-23.8) Mostre o resultado da inserção das seguintes palavras numa trie padrão: abab, baba, ccccc, bbaaaa, caa, bbaacc, cbcc, cbca.
9. (Goodrich e Tamassia R-23.9) Mostre o resultado das inserções das chaves do exercício R-23.8 numa trie compacta (radix tree).
10. O maior argumento para a utilização da representação compacta em árvores de sufixo (radix tree) é porque o pior caso para tries tradicionais requer espaço $O(m^2)$. Dê um exemplo desse pior caso e justifique porque ele requer espaço quadrático.
11. Como garantir que somente as folhas armazenem sufixos do padrão em uma árvore de sufixos? Justifique.
12. Sejam S_1 e S_2 duas strings arbitrárias. Um problema bastante comum envolvendo duas strings é determinar qual a maior substring compartilhada por ambas. Por exemplo, supondo que $S_1 = aauladealgoritmosefacil$ e $S_2 = osambaeoritmodocarnaval$ a maior substring compartilhada por ambas é *oritmo*. Na década de 1970, conjecturou-se que esse problema não possuía solução linear. Posteriormente, com o conhecimento das árvores de sufixo, desenvolveu-se um algoritmo linear (no tamanho total das strings) para encontrar a maior substring comum às duas. Descreva uma solução que resolva tal problema com a complexidade proposta.
13. (Adaptado de Goodrich e Tamassia R-23.11) Construa a árvore de sufixos para *cgtacgttcgtacg*. Qual a maior substring que se repete no texto?